

Ejercicio 1 Se consideran los siguientes programas:

```
var a : integer;
begin
  readln(a);
  case a of
    1: a:= a + 1;
    2: a:= 32;
    3,4: a:= a * 2;
  end;
  writeln(a);
end.
```

```
var a: integer;
begin
  readln(a);
  if a = 1 then
    a:= 1 + a;
  if (a >= 1) and (a <= 4) then
    if a = 2 then
      a:= 32
    else
      a:= a * 2;
  writeln(a);
end.
```

```
CONST N = 5;
TYPE Arreglo = ARRAY[1..N] OF Integer;
VAR a: Arreglo; i: 1..N;
```

a) for i:=1 to N do
if a[i] < a[i+1] then write (i, ' ');

b) for i:=1 to N do
if a[i-1] < a[i] then write (i, ' ');

c) for i:=1 to N do
if a[i] < a[i+1] then write (a[i], ' ');

d) for i:=1 to N-1 do
if i < i+1 then write (i, ' ');

e) for i:=1 to N-1 do
if a[i] < a[i+1] then write (i, ' ');

Respuesta: e

Indicar en qué casos ambos programas producen salidas distintas:

- a) Para cualquier valor que se ingrese.
- b) Solamente si se ingresa 1
- c) Solamente si se ingresa un valor mayor que 4.
- d) Solamente si se ingresa 3 o 4.
- e) Nunca producen salidas distintas cualquiera sea la entrada.

Respuesta: b

Ejercicio 2 Siendo i, j variables de tipo entero, indique cuál de los siguientes fragmentos de programa imprime 2 6 12.

1)

```
i:= 1; j:= 2;
while (i*j < 12) do
begin
  write (i*j, ' ');
  i:= j;
  j:= j+1;
end;
```

2)

```
j:= 2;
for i:= 1 to 3 do
begin
  write (i*j, ' ');
  j:= j+1;
end;
```

3)

```
i:=1; j:= 2;
repeat
  write (i*j, ' ');
  i:= j;
  j:= j+1;
until (i*j = 12);
```

- a) 1) y 2)
- b) 3)
- c) 2)
- d) 1) y 3)
- e) 2) y 3)

Respuesta: c

Ejercicio 4 Indicar cuál opción indica la salida del siguiente programa si se ingresa 123.

```
program fun (input, output);
var a, b: Integer;
function nose (x: Integer): Integer;
begin
  x:=x+1;
  nose:= x mod 10;
end;
begin
  read(a);
  b:= nose (a);
  writeln (a, ', ', b);
end.
```

- a) 124,12
- b) 124,4
- c) 123,3
- d) 123,4
- e) 123,5

Respuesta: d

Ejercicio 3 ¿Cuál de los siguientes fragmentos de programa escribe los índices i del arreglo a tales que a[i] < a[i+1]? Los tipos están declarados así:

Ejercicio 5 Sea el siguiente programa

```
program xyzw;
var
  xyz : boolean;
  i : integer;
begin
  xyz:= true;
  for i:= 1 to 7 do
  begin
    xyz:= not xyz or (i < 3);
    if xyz then
      writeln(i * 3, ' ')
    end
  end.
end.
```

Indique qué valores despliega:

- a) 3 6 9 12 15 18 21
- b) 3 6 12 18
- c) 9 15 21
- d) 3 6
- e) 3 9 15 21

Respuesta: b)

Ejercicio 6 Indicar cuál asignación es correcta para las declaraciones siguientes:

```
type
  Arreglo = array[1..10] of Char;

var
  a : Arreglo;
  c : Char;
  i : Integer;
  b : Boolean;
```

- a) $i := a[1] + a[2]$
- b) $c := a[1]$
- c) $a := a[1..10]$
- d) $b := a[1] > a[2]$
- e) $a[1] := 'a[2]'$

Respuesta: d

Ejercicio 7 Dadas las siguientes declaraciones:

```
const N = ...;

type
  Numeros = array[1..N] of Integer;

function todosPares(nums : Numeros): Boolean;
var i : Integer;
begin
  i := 1;
  while (i <= N) and (nums[i] mod 2 = 0) do
    i := i+1;
  todosPares := i > N
end;

var
  pares : Boolean;
  numLeidos : Numeros;
```

Decir cuál de los siguientes fragmentos de código compila correctamente (asuma que el arreglo numLeidos fue cargado con datos).

```
a)
if todosPares then
  writeln('Mensaje cualquiera')
```

```
b)
if pares := todosPares then
  writeln('Mensaje cualquiera')
```

```
c)
if pares := todosPares(numLeidos) then
  writeln('Mensaje cualquiera')
```

```
d)
if todosPares(numLeidos) then
  writeln('Mensaje cualquiera')
```

```
e)
todosPares(numLeidos);
if todosPares then
  writeln('Mensaje cualquiera')
```

Respuesta: d

Ejercicio 8 Sea una asignación válida en pascal:

```
a := b
```

Indicar cuál de las siguientes afirmaciones es verdadera:

- a) b puede no haber sido declarada.
- b) a y b deben tener el mismo tipo
- c) b no puede ser una constante
- d) a no puede ser una constante
- e) a puede ser de tipo integer y b de tipo real.

Respuesta: d

Ejercicio 9 Indique cuál de las siguientes declaraciones de tipos es sintácticamente correcta

- a) arreglo1 = array [100..1] of integer
- b) arreglo2 = array [1.0 .. 10.0] of integer
- c) arreglo3 = array ['1'..'10'] of char
- d) arreglo4 = array [-1..1] of array [char] of real
- e) arreglo5 = array [1..20] of 400

Respuesta: d

Ejercicio 10 Diga cuál de las siguientes afirmaciones, sobre el programa (incompleto) que se da a continuación, es verdadera:

```
procedure p(x,y: real);

  procedure q(x: real) : integer;
  var y: integer;
  begin
    ... (* sentencias de q *)
  end; {q}

  function f(x: real) : integer;

    function g(y: real) : real;
    begin
      ... (* sentencias de g *)
    end; {g}
  begin
    ... (* sentencias de f *)
  end; {f}
begin
  ... (* sentencias de p *)
end; {p}
```

- a) Los parámetros nominales de p son visibles en las sentencias de f.
- b) El parámetro nominal x de p es visible en las sentencias de g.
- c) La variable y local a q es visible en las sentencias de f.
- d) El parámetro nominal y de p es visible en las sentencias de f, pero no es visible ni en las sentencias de q ni en las de g.
- e) Si se define una variable local en f de nombre y, debe ser de un tipo distinto a real.

Respuesta: d

Ejercicio 11 Dadas las declaraciones:

```
var i : integer;
    a : array [1..5] of integer;
```

¿Cuáles de las siguientes condiciones nunca producen error en tiempo de ejecución, independientemente del valor de i? Asuma evaluación por circuito corto.

- 1) (i > 0) and (a[i] = 7)
- 2) (i > 0) or (i < 5) and (a[i+1] = 7)
- 3) (i > 0) and (i <= 5) and (a[i mod 5 + 1] = 7)
- 4) (i > 0) and (i < 6) or (a[i] = 7)
- 5) (i > 0) or (i < 6) or (a[i] = 7)

- a) 1) y 3)
- b) 2) y 4)
- c) 3) y 4)
- d) .3) y 5)
- e) 4) y 5)

Respuesta: d

Ejercicio 12 De los siguientes fragmentos de programa, indique cuáles son **todos** los que obtienen en la variable fact el factorial de un número n mayor que 0

Se recuerda la definición de factorial:

$$0! = 1$$

$$(n + 1)! = n! \times (n + 1)$$

```
var fact, i: Integer;

1)

fact := 0;
for i := 1 to N do
  fact := fact * i;
```

```
2)

fact := 1;
for i := 1 to N do
  fact := fact * i;
```

```
3)

fact := N;
for i := N-1 downto 1 do
  fact := fact * i;
```

```
4)

fact := 1;
for i := 1 to N div 2 do
  fact := fact * i * (N+1-i);
```

```
5)

fact := 1;
k := 0;
for i := 1 to N do
begin
  repeat
    k := k + 1
  until (k - i) mod 2 = 0;
  fact := fact * k
end
```

- a) 1)
- b) 2)
- c) 2) y 3)
- d) 2), 3) y 4)
- e) 2) y 5)

Respuesta: e

Ejercicio 13 Considere el siguiente fragmento:

```
if tambor(x,x,12) then
  writeln('no tambor')
else
  writeln(x)
```

Donde la variable x se supone declarada como integer.

¿Cuál de los siguientes podría ser un encabezado válido para la función tambor?:

```
a) function tambor(var x,y: integer;
  z: integer): integer;
```

```
b) function tambor(var x,y,z: real) : boolean;
```

```
c) function tambor(var x : integer;
  y,z: real) : boolean;
```

```
d) function tambor(var x,y: real;
  z : integer): boolean;
```

```
e) function tambor( x: real;
  var y,z: integer): boolean;
```

Respuesta: c

Ejercicio 14 ¿Cuál de las siguientes expresiones no puede ser asignada a una variable de tipo Real?

- a) 3 + 2.5
- b) trunc(3.5) + 2.5
- c) (4 mod 3) + trunc(3.5)
- d) round(4.5) > (3.5 * 2)
- e) 3.0 - 2.0

Respuesta: d

Ejercicio 15 Considere el siguiente código:

```
var valor, incremento, i : integer;
begin
  valor := 0;
  i := 1;
  read(incremento);
  repeat
    i := i + 1;
    valor := valor + incremento;
    read(incremento);
  until (incremento > 8) or (i > 3);
  writeln(valor);
end.
```

Suponiendo que la entrada es 3, 5, 6, 11, 2 decir cuál es el valor que se desplegará:

- a) 11
- b) 18
- c) 14
- d) 22
- e) 17

Respuesta: c

Ejercicio 16 Dadas las declaraciones:

```
var z,w : integer;

procedure p (var x : integer; y : integer);
begin
  x := y + z;
  z := x + z
end;
```

Suponga que z = 2 y w = 3. Indicar cuáles de las siguientes invocaciones dejan a z con valor 8.

- 1) p(z, z)
- 2) p(w, z)
- 3) p(z, w)
- 4) p(w, w+1)
- 5) p(z, z+1)

- a) 2) y 3)
- b) 2) y 4)
- c) 1) y 3)
- d) 1) y 4)
- e) 3) y 5)

Respuesta: d

Ejercicio 17 Para $n = 0$ y cualquier valor de m , indique cuál de las expresiones siguientes no da un error en tiempo de ejecución. Asuma evaluación por circuito corto.

- a) $(n = 0) \text{ and } (m \text{ div } n > 0)$
- b) $(n = 0) \text{ or } (m \text{ div } n > 0)$
- c) $(m \text{ div } n > 0) \text{ and } (n = 0)$
- d) $(n < 0) \text{ or } (m / n > m)$
- e) $(m \text{ div } n = m \text{ div } n) \text{ and } (n = 0)$

Respuesta: b

Ejercicio 18 Dado el siguiente fragmento de código:

```
for i := 0 to n do
  Write('*');
```

¿Cuál de las siguientes afirmaciones es verdadera?

- a) Si $n > 0$, se despliegan n asteriscos.
- b) Si $n < 0$, se produce un error en tiempo de ejecución.
- c) Si $n < 0$, la ejecución del ciclo no termina nunca.
- d) Si $n = 0$, no se despliega nada ni se produce ningún error.
- e) Si $n < 0$, no se despliega nada ni se produce ningún error.

Respuesta: e

Ejercicio 19 ¿Cuáles de los siguientes fragmentos de programa dejan el arreglo a inalterado?

```
const N = ...;

var a : array [1..N] of integer;
    i, aux : integer;

1)

for i := 1 to N do
  a[i] := a[N+1-i];
```

```
2)

for i := 1 to N do
  a[i] := a[(N+i-1) mod N + 1];
```

```
3)

for i := 1 to N div 2 do
  a[N+1-i] := a[i];
```

```
4)

for i := 1 to N div 2 do
begin
  aux := a[i];
  a[i] := a[i + N div 2];
  a[i + N div 2] := aux
end;
```

```
5)

for i := 1 to N do
begin
  aux := a[N+1-i];
  a[N+1-i] := a[i];
  a[i] := aux
end;
```

- a) 1) y 2)
- b) 2) y 3)
- c) 2) y 5)
- d) 3) y 4)
- e) 3) y 5)

Respuesta: c

Ejercicio 20 Se considera la siguiente definición:

```
var a : Array [1..MAX] of Integer;
    i, res : Integer;
```

Indicar cuál de los siguientes fragmentos de programa deja en la variable res el menor valor k tal que:

- k es par
- $a[k]$ es impar

En caso de no existir tal valor, deja 0 en la variable res .

```
a)
res:=0; i:=2;
while (i <= MAX) and
  (a[i] mod 2 = 0) do
  i:=i+2;
if (i <= MAX) then res:=i;
```

```
b)
res:=0; i:=1;
while (i <= MAX) and
  (a[i] mod 2 <> 0) do
  i:=i+2;
if (i <= MAX) then res:=a[i];
```

```
c)
res:=0; i:=2;
while (a[i] mod 2 = 0) and
  (i <= MAX) do
  i:=i+2;
if (i <= MAX) then res:=i;
```

```
d)
res:=0; i:=1;
while (a[i] mod 2 <> 0) and
  (i <= MAX) do
  i:=i+2;
if (i <= MAX) then res:=a[i];
```

```
e)
res:=0; i:=0;
while (i <= MAX) and
  (a[i] mod 2 = 0) do
  i:=i+2;
if (i <= MAX) then res:=i;
```

Respuesta: a